Homework Assignment 1

ECE438

Due 2/21/2020

1. **(10 points)** Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor.

2. **(10 points)** The following questions examine die yield and cost using data provided by Intel along with industry assumptions.

   (a) The Intel Core i7 die in a 45nm process technology node is 18.9mm by 13.6mm. Calculate the number of dies per 300mm wafer.

   (b) Calculate the die yield assuming 0.020 defects per square centimeter and a process-complexity factor, $N = 7.0$. Also assume 99% wafer yield.

   (c) Assuming the cost of processing a single 300mm wafer is $6,500, calculate the cost per die of a Core i7.

   (d) Price is used to describe what the consumer would pay for a chip, whereas cost in this case describes what Intel would pay for the raw material and processing. Assuming Intel could sell each fully functioning die for 1.5x the cost calculated in the previous question, how much money per wafer would Intel gain?

   (e) The Core i7 contains 4 cores. If 65% of the failed dies could be repackaged as dual-core processors and sold for 60% of what the quad-core processors sell for, how much money per wafer would Intel gain? If 15% of their total gains go to R&D, how much is left over?

3. **(10 points)** Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 1.8 GHz clock rate with a CPI of 0.8. P2 has a 2.4 GHz clock rate with a CPI of 1.2. P3 has a 3.2 GHz clock rate with a CPI of 1.6.

   (a) Which processor has the highest clock rate? Calculate the clock period of each processor.

   (b) Calculate the Million Instruction Per Second (MIPS) of each processor. Which one has the highest MIPS rating?

   (c) The same benchmark is compiled for each of the processors such that P2 executes 19 million instructions to completion (i.e. has a dynamic instruction count of 19 million instructions), while P1 and P3 have a dynamic instruction count of 22.0 million instructions. Calculate the execution time of each processor? Which processor is fastest?

   (d) Assuming the reference machine takes 35 milliseconds to execute the given benchmark, calculate the speedup of each processor compared to the reference machine.

4. **(10 points)** Applications are often times categorized as either computation bound or I/O bound. The former spends most of its execution time on computation, while the latter spends most of its time waiting for I/O (i.e. disk, network access, etc.). Use Amdahl's Law to answer the following questions:

   (a) Suppose you have an application that spends 60% of its time waiting for disk access and 40% of its time on computation. Is this system I/O bound or computation bound?

(b) If you were to run this application on a processor that is 20x faster but with the same I/O speed, what would be the potential speedup?

(c) If instead, you ran the application on a system with the same processor speed but a disk that is 3x faster, what would be the potential speedup?

(d) Describe how you might go about improving the disk access of this system.

5. **(10 points)** For your research, you profiled an application that requires 1000 seconds of execution time on a single-core architecture. You would like to design a multi-core architecture to speed up this particular application. Use Amdahl's Law to answer the following questions:

   (a) If you were to rewrite the application to take advantage of a multi-core architecture, what would be the maximum achievable speedup on an architecture with four cores? What would the new execution time be?
   Assuming 800 seconds of the application's execution time is perfectly parallelizable, while 200 seconds must be executed sequentially. Also assume the performance of each core in the quad-core architecture equals that of the original single-core architecture.

   (b) What would be the execution time and speedup achievable with an eight-core machine?

   (c) What would be the maximum speedup achievable on a machine with unlimited cores?

   (d) What would be the maximum speedup achievable on a machine with unlimited cores if 90% of the execution time could be perfectly parallelized?

6. **(10 points)** A cryptographic operation takes 2 seconds to run on a simple embedded processor core. You are considering the design of a coprocessor to accelerate this cryptographic operation in order to improve the energy efficiency of the system. Answer the following questions, using Amdahl's Law where appropriate.

   (a) If 95% of the execution time can be accelerated with a coprocessor, what is the maximum speedup theoretically achievable, assuming no computation overlap between the accelerator and the processor?

   (b) If you were able to design an accelerator that can execute the aforementioned 95% execution time with a 50x speedup, what would be the maximum achievable speedup of the entire cryptographic operation? What is the new execution time?

   (c) What is the energy improvement of a cryptographic operation with the coprocessor discussed in the previous question if the power of the system doubles with the addition of the accelerator?

   (d) Plot the speedup of the overall cryptographic operation as the coprocessor speedup varies from 1x to 50x assuming 95% of the execution time can be executed on the accelerator. In the same figure, plot the speedup assuming 80%, 85%, and 90% of the execution time can be executed on the accelerator. Be sure to properly label the axis of your graph and use a legend to differentiate between the four different lines. You may use any graphing tool of your choice.

7. **(15 points)** Imagine you designed a small embedded processor and synthesized it for a 90nm process technology node with a target clock rate of 500MHz. During power analysis, you found that at the target clock rate with a supply voltage of 1.2V, this processor draws 1.0W of dynamic power and 0.2W of static power. Consider the following power and energy trade-offs:

   (a) Assuming a cryptographic operation takes 1 second to complete on your processor, what is the energy per cryptographic operation at the target clock rate?

   (b) For certain applications, your processor performs cryptographic operations 2x faster than necessary. If you were to slow the clock down to 250MHz without adjusting the voltage, what would be the energy per cryptographic operation? What would be the overall power draw?

(c) Assuming you could safely drop the voltage to 0.9V when operating at a 250MHz clock, recalculate the power draw and energy per cryptographic operation. Assume the leakage *current* remains the same.

(d) Static timing analysis shows that if you were to increase your supply voltage to 1.4V, you could run your processor at 650MHz. Recalculate the power draw with the increased voltage and frequency parameters. Recalculate the energy per cryptographic operation.

(e) The above questions looked at how the power and energy can be changed by varying the clock rate and voltage of a processor. Modern processors change these parameters dynamically, using a technique called Dynamic Voltage Frequency Scaling (DVFS). What are some other techniques for reducing energy? Briefly describe these techniques.

(f) The technology node that a particular processor is fabricated with can also affect the energy efficiency. Imagine that you resynthesized your processor at a 130nm process technology node with a 1.5V supply voltage. In order to maintain the same transistor count, you set your target clock rate for 333MHz. Assuming the leakage *current* remains the same and that the capacitance of the design approximately scales linearly with the feature size, calculate the dynamic and static power for your processor at the 130nm node. What is the energy per cryptographic operation at the 130nm node and how does this compare to that of the 90nm node?

8. **(10 points)** The execution times of a few SPEC2006Cint benchmarks on three different machines are shown in the table below.

Benchmark	Ultra 5 Time (sec)	Machine A Time (sec)	Machine A SPEC ratio	Machine B Time (sec)	Machine B SPEC ratio
perlbench	9,770	454		253	
bzip2	9,650	520		438	
gcc	8,050	461		234	
mcf	9,120	268		150	
gobmk	10,490	429		383	
hmmer	9,330	197		135	
sjeng	12,100	529		362	
libquantum	20,720	91		7	
h264ref	22,130	599		427	
omnetpp	6,250	305		163	
astar	7,020	327		234	
xalancbmk	6,900	253		117	
Geometric mean					

(a) Using the Ultra 5 as a baseline, fill in the SPECratio columns above for Machines A and B.

(b) Compute the geometric mean of the SPECratios for both machines.

(c) Given the benchmarks above, which machine (A or B) is faster? By how much?

(d) libquantum has fallen victim to substantial compiler optimizations as evidenced by such extraordinary speedups over the reference machine. Consequently, it has been removed from SPEC2017. Recalculate the geometric mean of the normalized performance without libquantum. Compare machine A to machine B excluding libquantum.

9. **(10 points)** Compilers can have a profound impact on the performance of an application. The following questions examine the relationship between dynamic instruction count, CPI, and clock rate in the context of two different compilers:

(a) Assume that for a given program, compiler A results in a dynamic instruction count of $8 * 10^8$ and has an execution time of 1.1 seconds, while compiler B results in a dynamic instruction

count of $1.2*10^9$ and an execution time of 1.6 seconds. Find the average CPI for each program given that the processor has a clock cycle time of 1ns.

(b) Assume the compiled programs run on two different processors with different clock rates. If the execution times on the two processors (one running compiler A's code and the other running compiler B's code) are the same, what is the ratio of the clock rate of the processor running compiler A's code over the clock rate of the processor running compiler B's code? Use the dynamic instruction count and calculated CPI from the previous part.

(c) A new compiler is developed that has a dynamic instruction count of only $7.5*10^8$ instructions and has an average CPI of 1.3. What is the speedup of using this new compiler versus using compiler A or B on the original processor?